CVWO AY21/22 Assignment – Final Submission **Lee Jia Wei (A0235256E)**

Repository: https://github.com/Beebeeoii/do-gether

Main URL: https://do-gether.jiaweilee.com

Reflections:

This assignment has been a fun and fruitful one despite there being many instances during this journey when I felt overwhelmed with handling both the frontend and backend. Although I have worked with both frontend and backend development before, React for frontend development and Golang for backend development were new to me. I saw this as a learning opportunity to expose myself to new technologies, architectures and frameworks.

Frontend

Personally, I found frontend development the most challenging. In addition to managing the business logic behind the application, I had to ensure that states are properly managed and that views are properly rendered. Though I have worked with Angular for frontend development before, React + Redux was unique and I spent quite some time initially trying to wrap my head around the entire flow of data with a Redux store being data's 'source of truth'.

Furthermore, I had many mind-boggling sessions with trying to style the webpage to look like the one I have in my mock-up. This is mainly due to my poor command of CSS and its properties. However, I am glad that I persevered through those challenging and possibly very annoying periods as I have picked up much knowledge and am now more confident.

Backend

I enjoyed developing the backend as it gave me a "sense of control" over the entire application. However, with great power comes great responsibilities. With the backend mediating between clients and the database, I had to ensure that restricted operations (e.g. accessing another user's to-do lists and tasks without permissions) are denied. I was able to better appreciate how important the backend is in ensuring that operations and data being executed, written or read by users are properly controlled.

I vividly remember thinking through the issue of access control of a user's to-do tasks. With the feature of friends, users can send, decline and accept friend requests to/from another user on the platform and invite them to a to-do list that they have created so that both the user and the invited friend are able to view, create, edit and delete the tasks in the list. There were several cases which I must consider thoroughly to ensure that no edge cases are left.

I chose to work with Golang instead of Ruby because I wanted to gain an even deeper understanding of Go. Though I have worked briefly with both languages before, I am enticed by the uniqueness of Go as a language with both the convenience of higher-level programming languages and the amount of control available to lower-level programming (though incomparable to languages like C or Rust).

Database

It is my first time working with a SQL database. It was fruitful and insightful as I learnt how relational databases work and managed to gain a much better understanding on when it would be more beneficial to utilise a relational database (e.g. PostgreSQL) and when to use a NoSQL database (e.g. MongoDB).

Having only worked with NoSQL databases before, I am now able to appreciate the options developers have when it comes to choosing the most suitable data management choice available for each unique use case.

Deployment

Docker

I felt most uncomfortable at this stage as it was my first time working with Dockerfiles and docker compose. Although I have used Docker before and vaguely understood how Docker works, creating my first Docker images and composing them posed some technical questions initially (making the containers communicate with one another, exposing of ports etc). Containerising is very useful, especially in scaling and orchestration of services and I am glad that I challenged myself to explore this uncomfortable, yet very rewarding route.

Hosting

I chose to host the application on a bare metal VPS instead of a managed service. I wanted to utilise the Docker images that I have created and deploy them as containers. Initially, I explored what Heroku can offer but unfortunately, their free tier only allowed up to 2 dynos but I require 3 as there are 3 containers (frontend, backend and database) being composed together. I considered the alternative of hosting the frontend server on its own via a frontend managed service like Vercel or Netlify. However, that would not enable me to fully experience how a docker-composed application works (as the frontend server is now being hosted on its own). Although hosting the application on a VPS encompasses more work to be done (setting up of SSL, reverse proxy etc), I am able to maintain full control over the application and it enabled me to experience first-hand how servers actually work.

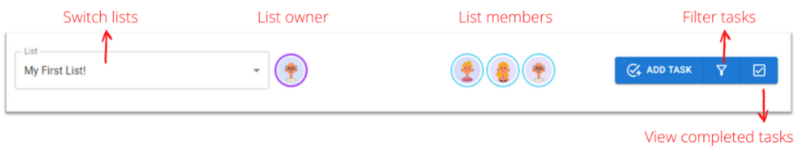
Summary

This project has been a meaningful one as I experienced the planning and execution processes. It was undoubtedly challenging at times as I sought to understand and appreciate new technologies, architectures and best practices. Having to manage both the frontend and backend, and tying them together required thoughtful code organisation and structure. As software developers, we are never developing just for ourselves, but also for others and with others. This project highlighted the importance of code clarity as the codebase grows with the features being implemented. Having a modular and clear code is truly important.

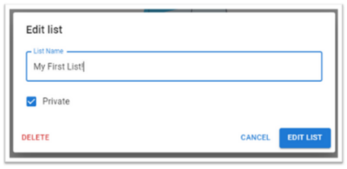
Although the application is far from perfect and there are more that I wish to improve on but am unable to as school and other commitments kick in, it is something I am truly proud of and would like to continue working on it during my spare time.

User Manual:

Dashboard: List/Task Tools



Edit list

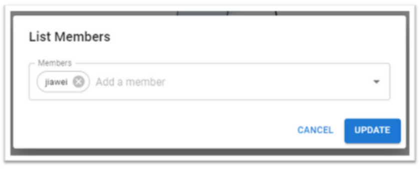

Edit list

List Name: My First List!

☑ Private

DELETE CANCEL EDIT LIST

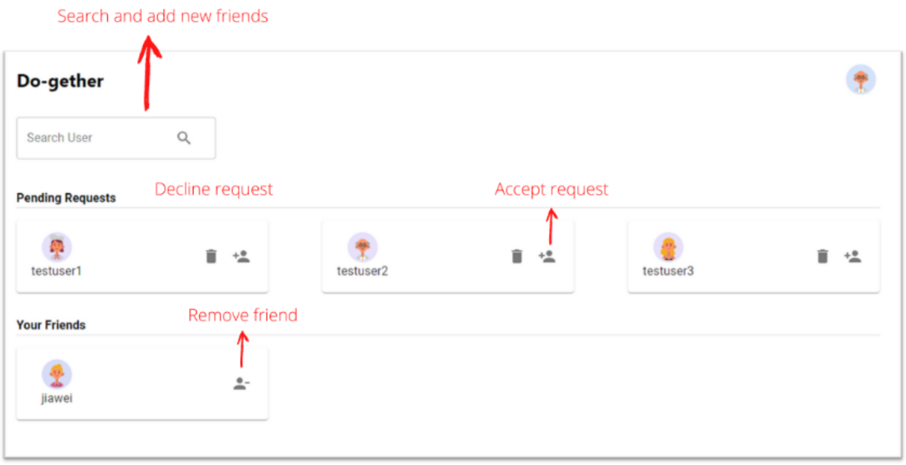
Add Members to list


List Members

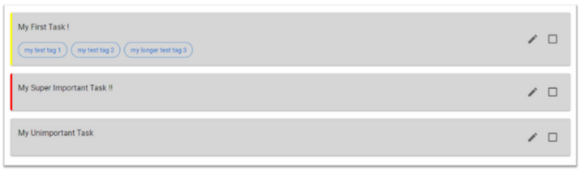
Members: jiawei Add a member

CANCEL UPDATE

Friends



Dashboard: List/Task Tools

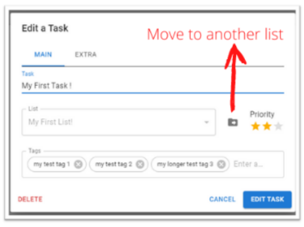

My First Task !

my test tag 1 | my test tag 2 | my longer test tag 3

My Super Important Task !!

My Unimportant Task

Drag and drop tasks around to reorder them (order is saved)



Edit a Task

MAIN EXTRA

Task: My First Task !

List: My First List!

Priority: ★★☆

Tags: my test tag 1 | my test tag 2 | my longer test tag 3 | Enter a...

DELETE CANCEL EDIT TASK

Edit Task